

Генерация наборов ключей для профилирования

Алгоритм генерации наборов ключей основан на использовании сочетаний. В комбинаторике сочетанием из n по k называется набор из k элементов, выбранных из данного множества, содержащего n элементов. Наборы, которые отличаются исключительно порядком следования элементов (но не составом), считаются одинаковыми.

Алгоритм перебора сочетаний

Для перебора сочетаний выберем удобное стандартное представление сочетаний, например в виде монотонно возрастающего набора из m чисел, лежащих в диапазоне $1:n$ и будем перебирать только такие стандартные записи сочетаний.

Имеем массив x_1, \dots, x_m номеров, включенных в сочетание.

Начальное состояние: Принять $x_i = i$ для всех i , принадлежащих множеству $1:m$.

Стандартный шаг: Просматривать компоненты вектора x , начиная с x_m , и искать первую компоненту, которую можно увеличить (нельзя увеличить $x_m = n$, $x_m - 1 = n-1, \dots$). Если такой компоненты не найдется, закончить процесс. В противном случае, пусть k — наибольшее число, для которого $x_k < n + m - k$. Увеличить x_k на единицу, а для всех следующих за k -ой компонентой продолжить натуральный ряд от нового значения x_k , т.е. положить $x_i = x_k + (i - k)$ для $i > k$.

Реализация алгоритма

Алгоритм перебора сочетаний был реализован с использованием мультипарадигмального языка программирования Scala.

```
package com.berezka

import java.io.PrintWriter
import scala.io.Source

object Main extends App {
  def loadKeysFromFile(path: String): List[String] = {
    Source.fromFile(path).getLines().toList
  }

  def saveToFile(keys: List[String])(implicit pw: PrintWriter): Unit = {
    pw.write(keys.toString().drop(5).dropRight(1).replaceAll(", ", ""))
    pw.write(" ")
    pw.write("\n")
  }

  override def main(args: Array[String]): Unit = {
    implicit val pw: PrintWriter = new
    PrintWriter(Option(args(1)).getOrElse("result.txt"))

    val keys: List[String] =
    loadKeysFromFile(Option(args(0)).getOrElse("keys.txt"))

    println((4 to 7).flatMap(keys.combinations).size)

    (4 to 7).flatMap(keys.combinations).foreach(saveToFile)

    pw.close()
  }
}
```

Данный алгоритм генерирует все возможные сочетания наборов ключей размером от 4 до 7 элементов.

Результаты работы

На вход алгоритму был передан файл со следующими ключами:

```
-mno-shared  
-mfp64  
-mfp64  
-mhard-float  
-msoft-float  
-msingle-float  
-mdouble-float  
-mabs=2008  
-mnan=2008  
-mllsc  
-mdspr2  
-mpaired-single -mips3d  
-mno-abicalls -msym32  
-mno-gpopt -mno-local-sdata -mno-extern-sdata  
-membedded-data -muninit-const-in-rodata  
-mexplicit-relocs  
-mload-store-pairs
```

В результате работы алгоритма было сгенерировано 40392 сочетаний ключей.